# REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE 31 Jan 1996 | 3. REPORT TYPE AND DATES COVERED Final Technical Report, 01 Dec 91 to 30 Sept 95 |
|---|---|---|

**4. TITLE AND SUBTITLE**
Connectionist modeling of basal ganglia motor circuitry

**5. FUNDING NUMBERS**
ONR G N-00014-92-J-1132

**6. AUTHOR(S)**
Garrett E. Alexander

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Dept. of Neurology, WMB 6000
Emory University School of Medicine
1639 Pierce Drive, P.O. Drawer V
Atlanta, GA 30322

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Dr. Thomas McKenna
Office of Naval Research
Code 342 CN
Ballston Tower One, 800 North Quincy Street
Arlington, VA 22217-5660

**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION / AVAILABILITY STATEMENT**
Approved for public release; distribution unlimited

DTIC QUALITY INSPECTED 4

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** (Maximum 200 words)

Abstract: Using a self-organizing, topology-preserving, sensorimotor architecture, we developed two types of neural networks that were capable of learning, without supervision, to control a simulated, three-segment robot arm with variable degrees of freedom (3,4 or 6 df). One type was an endpoint or posture-controlling network, and the other was a trajectory controller. The hidden layers in these networks consisted of both 2D and 3D lattices comprising from 729 to 1728 neurons. Through a process of trial and error, all networks learned to control the positioning of the distal end of the robot arm within a 3D workspace. The workspace was either a hemisphere or a cube centered at the origin of the stimulated limb. When tested after training that ranged from 2000 to 12000 trials, both networks achieved relatively uniform placement accuracy throughout the workspace, the level of accuracy varying directly with the number of processing elements and asymptotically with the duration of training. The number of trials required to achieve maximum accuracy was approximately 5 times the number of neurons in the hidden layers.

**14. SUBJECT TERMS**
Connectionist, neural networks, motor control

**15. NUMBER OF PAGES**
8

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | Unlimited |

19960213 036

## ONR FINAL TECHNICAL REPORT

Project Title:   Connectionist Modeling of Basal Ganglia Motor Circuitry

Project Dates:   12/01/91 to 09/30/95

Principal Investigator Name:   Garrett E. Alexander, M.D., Ph.D.

Institution:   Emory University School of Medicine, Atlanta, GA  30322

Abstract:       Using a self-organizing, topology-preserving, sensorimotor architecture, we developed two types of neural networks that were capable of learning, without supervision, to control a simulated, three-segment robot arm with variable degrees of freedom (3, 4 or 6 df). One type was an endpoint or posture-controlling network, and the other was a trajectory controller. The hidden layers in these networks consisted of both 2D and 3D lattices comprising from 729 to 1728 neurons. Through a process of trial and error, all networks learned to control the positioning of the distal end of the robot arm within a 3D workspace. The workspace was either a hemisphere or a cube centered at the origin of the simulated limb. When tested after training that ranged from 2000 to 12000 trials, both networks achieved relatively uniform placement accuracy throughout the workspace, the level of accuracy varying directly with the number of processing elements and asymptotically with the duration of training. The number of trials required to achieve maximum accuracy was approximately 5 times the number of neurons in the hidden layers.

During the initial phase of the project, we developed a new type of neural network architecture that is capable of learning a broad range of sensorimotor transformations without the necessity of supervised learning and without resorting to the inclusion of ad hoc, error-correcting algorithms. The new architecture represents a modification of the self-organizing sensory maps that Kohonen and others have developed as potential models for many types of biological sensory circuits (and as effective sensory analyzers in their own right.) The modified version extends the principle of a topography-generating *neighborhood function* from its original application in generating topographic sensory maps to a more generalized implementation that makes it possible to generate topographic motor maps as well.

As with the Kohonen-style sensory maps, the basic architecture consists of a self-organizing, 2- or 3-dimensional lattice of simulated neurons whose synaptic *input* weights are adjusted according to a Hebb-like learning rule that emphasizes *presynaptic* information and serves to bring the population of neurons to a state in which it represents a topographic map of the input space. The new variation which we have developed requires the synaptic *output* weights of these same neurons to be controlled by a learning rule that in most respects is analogous to that used for the input weights, with the critical exception that the output rule places special emphasis on *postsynaptic* activity levels.

The new sensorimotor architecture we have developed does require effective normalization of weights as well as of input vectors. From a biological standpoint, weight normalization has often been justified on the basis of evidence suggesting the existence of homeostatic mechanisms that maintain a roughly constant number of synaptic contacts impinging on a given neuron (with reinnervation compensating for functional or pathological denervation), or because of fixed limitations in the metabolic machinery available to support synaptic processing. Biological justification for the normalization of inputs has been provided by Grossberg's mathematical demonstration that this can be accomplished by a suitable combination of lateral inhibition and self-excitation, since this form of local circuit organization is present almost universally throughout the central nervous system. The same type of local circuit organization can also account for both the competitive, winner-take-all activation process and the neighborhood learning function required for the generation of self-organizing maps.

The project was aimed at the eventual development of neural network models of basal ganglia motor circuitry that could learn to control the target-seeking movements of a simulated, 3-segment robot arm with redundant degrees of freedom. The motor output circuitry employed in our new models consist of pairs of agonist-antagonist neurons, each of which controls one of the arm's degrees of freedom, in analogy with motor neuron pools controlling antagonistic muscle pairs. The self-organizing lattice of neurons that perform the sensorimotor transformation is fully connected to the output circuitry. In the course of learning, each neuron in the sensorimotor lattice acquires a set of output weights (connecting it to the ensemble of agonist-antagonist pairs of output neurons) that leads to the generation of a movement appropriate for the sensory vector to which that neuron is tuned. During the same learning process, each such neuron also acquires a set of input weights that determines its sensory tuning. Because of the topology-conserving nature of the lattice organization, in combination with the neighborhood function that arises from appropriate lateral and reflexive local circuit connections, neighboring sensorimotor neurons acquire sensory as well as motor properties that are similar in proportion to their proximity to one another. This results in the formation of a topographic sensorimotor map.

One difficulty with many types of Kohonen-style self-organizing sensory maps is that their performance may be limited by the effects of vector quantization. That is, the number of distinct outputs that the network can generate is usually limited by the number of neurons comprising the map. Previous efforts to extend such maps to the service of motor control problems have tried to deal with this problem of discretization by including hybrid structures that permit interpolation between the designated outputs of neighboring neurons when the actual input vector falls somewhere between the respective input vectors to which the neurons are tuned. For example, Ritter and colleagues have used Jacobian matrices, assigned to each processing element in the control lattice, to effect a form of interpolation. While effective in dealing with the problem of discretization, however, such ad hoc strategies have generally not involved solutions that are biologically plausible.

The new architecture that we have developed deals with the discretization problem in a generalized way that is also consistent with the constraints suggested by neurobiology. This is accomplished by permitting the output to be generated by a *cluster* of neurons surrounding and including the "winning" neuron (whose input weights are most closely tuned to the current input vector), rather than following the usual strategy of letting only the winning unit determine the output. Further advantages of the new architecture for the modeling of brain circuitry include its

2

inherent topographic coding of input and output variables (permitting ready comparisons with the topographic mappings in biological motor circuits), its conformance with known neurobiological constraints, and its capacity to permit independent as well as coordinated control of motor output by any number of input variables (e.g., target location, motor error, desired limb configuration, desired hand or joint velocity). This last feature is readily implemented by the convergence of separate sensory maps (representing, for example, separate cortical sensory areas, each processing a different set of input variables) onto a common sensorimotor lattice (representing a sensorimotor nodal point, such as the putamen or primary motor cortex) that carries out the resultant sensorimotor transformation. It is possible that this type of architecture may eventually permit the development of functional models with some of the same versatility in behavioral control that is evident in mammalian motor systems.

The self-organizing, topology-preserving, sensorimotor architecture described above was used to develop and test two types of neural networks that proved to be capable of learning, without supervision, to control a simulated, three-segment robot arm with variable degrees of freedom (3, 4 or 6 df). One type was an endpoint or posture-controlling network, and the other was a trajectory controller. Networks were tested for their capacity to learn, without supervision, to control a simulated, three-segment robot arm with variable degrees of freedom (3, 4 or 6 df). The hidden layers in these networks consisted of both 2D and 3D lattices comprising from 729 (e.g., 27x27 or 9x9x9) to 1728 (12x12x12) neurons. Through a process of trial and error, all networks learned to control the positioning of the distal end of the robot arm within a 3D workspace. The workspace was either a hemisphere or a cube centered at the origin (i.e., the shoulder joint) of the simulated limb. When tested after training that ranged from 2000 to 12000 trials, all networks achieved relatively uniform placement accuracy throughout the workspace, the level of accuracy varying directly with the number of processing elements and asymptotically with the duration of training. The number of trials required to achieve maximum accuracy was approximately 5 times the number of neurons in the hidden layers.

Each type of network was tested in two fundamentally different configurations: a) as a single-step *posture* controller, with the network learning to associate desired target coordinates (for placement of the hand) as inputs with appropriate final joint angles (absolute angles) as outputs, and b) as a multi-step *trajectory* controller, with the network learning to associate desired target coordinates plus current joint angles as inputs with appropriate *changes* in joint angles (delta angles) as outputs. Like the various "equilibrium point" models of skeletomotor control, our posture controller networks do not require an estimate of current motor error to compute an appropriate motor response (they only need to know the target or goal of the movement), nor do they control the specific trajectory the limb will follow in acquiring the target. These networks simply learn to select a new posture for the limb that places the hand at the desired target coordinates. In contrast, our trajectory controller networks do require an estimate of current motor error (implicit in the combination of target and proprioceptive inputs) in order to select an appropriate motor output, and the resulting output command does specify an explicit trajectory for the limb to follow.

All of these networks were successful in learning to track targets located anywhere in the accessible workspace. Somewhat surprisingly, we found that neither the rate of learning nor the final accuracy of tracking seemed to depend on the degree of redundancy in the robot limb. It appeared to be no more difficult for these networks to control a limb with 6 df than one with 3 df. For networks of comparable size, there were also no discernible differences in performance between those whose neurons were arranged in two dimensions (like cortex) and those with three dimensional architectures (like subcortical nuclei), even though learning was based on neighborhood functions and local cooperativity among neurons. All networks showed clear and striking topographical mappings in their representations of the input space. This was also the case for the representations of the output space by the trajectory controllers. For the posture controllers, however, the topography of the output representations was highly "fractured", due to the underdetermination of input-output mappings with this configuration. Another salient difference between posture controllers and trajectory controllers was that the latter needed to be significantly larger in order to achieve comparable accuracy, presumably because of the necessity that target-to-output mappings be of the one-to-many variety for trajectory controllers (to allow for different starting postures), while one-to-one mappings were adequate for the posture controllers.

The endpoint or posture-controlling networks received visual information about target location and learned, through trial and error, to command limb postures that would place the hand in contact with the target. One of the keys to the success of these models was the use of Hebbian synapses that were specialized for either sensory or motor circuitry. The weights of adaptive synapses conveying sensory information were adjusted principally accordance with presynaptic activity levels, while adaptive synapses conveying motor information were adjusted mainly in

proportion to postsynaptic activity. To simulate the redundancy of biological limbs, the robot limb had three segments and up to six degrees of freedom of joint displacement. **Figure 1** illustrates the performance of a typical endpoint-controlling network after a period of training. This network consisted of 729 hidden neurons organized in a 9x9x9 three-dimensional lattice. After 4000 learning trials, the network was able to position the hand accurately throughout the workspace. In this example, the network was controlling a limb with six degrees of freedom (two at each joint). In this type of network, optimum accuracy varied directly with the number of hidden neurons, and was asymptotically related to the length of training. The number of trials required to achieve maximum accuracy was approximately five times the number of neurons in the hidden layers.

The use of a Kohonen-style neighborhood function insured cooperativity among neighboring neurons. This topology-preserving feature resulted in uniform representation of the workspace by the sensorimotor network. The manner in which the workspace came to be represented by the input weights of the hidden lattice of neurons is illustrated in **Figure 2**. The left-hand panel shows the distribution of input weights following the initial randomizaton. At this point there is simply random clustering of representations centered around the origin (0, 0, 0). The center panel shows the distribution of input weights after the network has been trained in a hemispheric workspace. That is, training targets were presented randomly but only within the hemisphere aligned along the positive x axis (radius = 5 units). The resulting distribution of input weights for the hidden lattice of sensorimotor neurons formed a relatively uniform representation of that hemispheric workspace. Similarly, the right-hand panel shows a cubic distribution of input weights when a cubic workspace (x = +/- 2.5, y = +/- 2.5, z = +/- 2.5) was used during training. In these networks, the output weights directly represented joint displacements rather than workspace coordinates. However, those joint displacements directly determined the resulting hand position. When translated into the corresponding hand positions, the output weights formed representations of the workspace that were very similar to those of the input weights.

We have also developed Hebbian trajectory-controlling networks. Whereas the endpoint-controlling networks generated new terminal postures in a single step by specifying absolute joint angles as their output commands, the outputs of the trajectory-controlling networks represented a series of *changes* in joint angles that specified a definite, multi-step trajectory that depended on the initial posture of the limb at the time the target or goal of the movement was presented. While the endpoint- or posture-controlling networks do not require proprioceptive input to guide their movements, the trajectory-controlling networks require both visual and proprioceptive inputs. Like the various "equilibrium point" models of skeletomotor control, our posture controller networks do not require an estimate of current motor error to compute an appropriate motor response (they only need to know the target or goal of the movement), nor do they control the specific trajectory the limb will follow in acquiring the target. These networks simply learn to select a new posture for the limb that places the hand at the desired target coordinates. In contrast, our trajectory controller networks do require an estimate of current motor error (implicit in the combination of target and proprioceptive inputs) in order to select an appropriate motor output, and the resulting series of output commands does specify an explicit trajectory for the limb to follow.
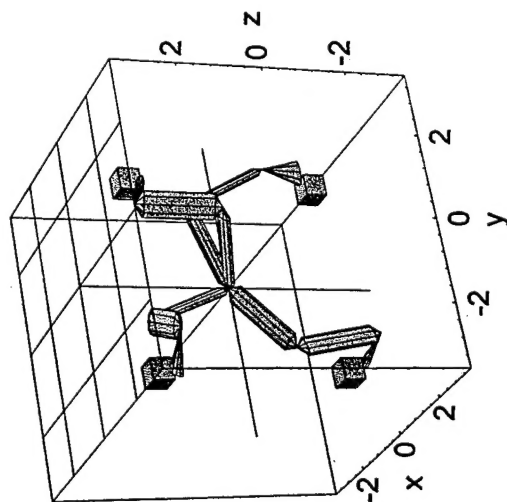
**Figure 3** illustrates the performance of a trajectory-controlling network that has learned to control the targeted hand movements of a limb with six degrees of freedom. This problem is substantially more difficult than that faced by the endpoint-controlling networks, even when the limb has comparable degrees of freedom. The endpoint controller can ignore the initial posture and simply associate a new terminal posture with the desired target location. The trajectory controller, however, must find approximate solutions to the targeting problem that are appropriate from all possible initial joint configurations. Remarkably, it is able to do so even with a relatively small sensorimotor lattice, as illustrated in this figure. This network consisted of 1728 hidden neurons organized in a 12x12x12 lattice. The performance illustrated here was obtained after only 6000 training trials. Due to the topology-preserving nature of the learning algorithm, trajectories associated with similar targets and initial postures tended to be comparable for a given network.

Both types of networks, endpoint (posture) controllers and trajectory controllers, were successful in learning to track targets located anywhere in the accessible workspace. Somewhat surprisingly, we found that neither the rate of learning nor the final accuracy of tracking seemed to depend on the degree of redundancy in the robot limb. It required no more learning for these networks to control a limb with six degrees of freedom than one with three. For networks of comparable size, there were also no discernible differences in performance between those whose neurons were arranged in two dimensions (like cortex) and those with three dimensional architectures (like subcortical nuclei),
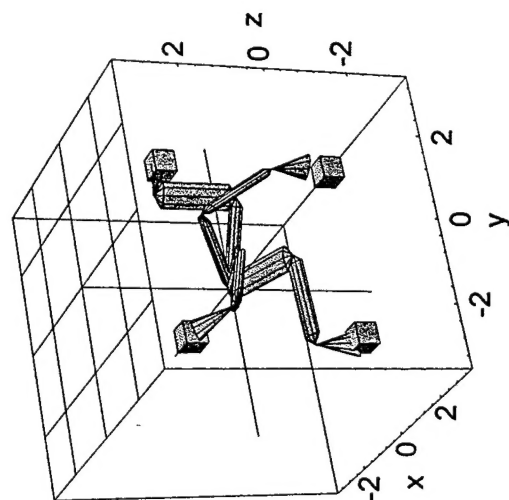
4

even though learning was based on neighborhood functions and local cooperativity among neurons. All networks showed clear and striking topographical mappings in their representations of the input space. This was also the case for the representations of the output space by the trajectory controllers. For the posture controllers, however, the topography of the output representations was highly "fractured", due to the underdetermination of input-output mappings with this configuration. Another salient difference between posture controllers and trajectory controllers was that the latter needed to be significantly larger in order to achieve comparable accuracy, presumably because of the necessity that target-to-output mappings be of the one-to-many variety for trajectory controllers (to allow for different starting postures), while one-to-one mappings were adequate for the posture controllers.

Hebbian End Point (Posture) Controller: Testing on New Targets after Training in Hemispheric Workspace

Sample of 4 Distal Targets

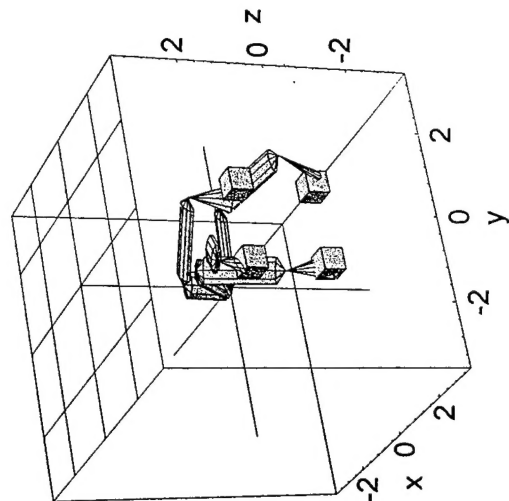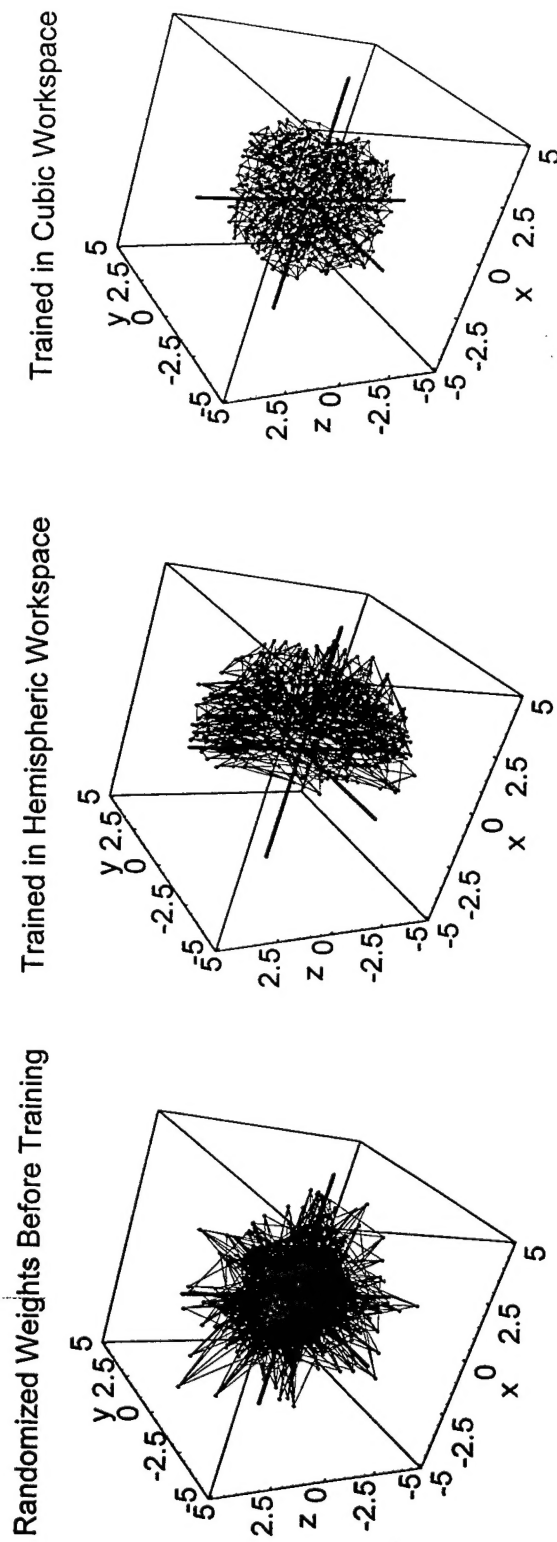Sample of 4 Mid-Range Targets

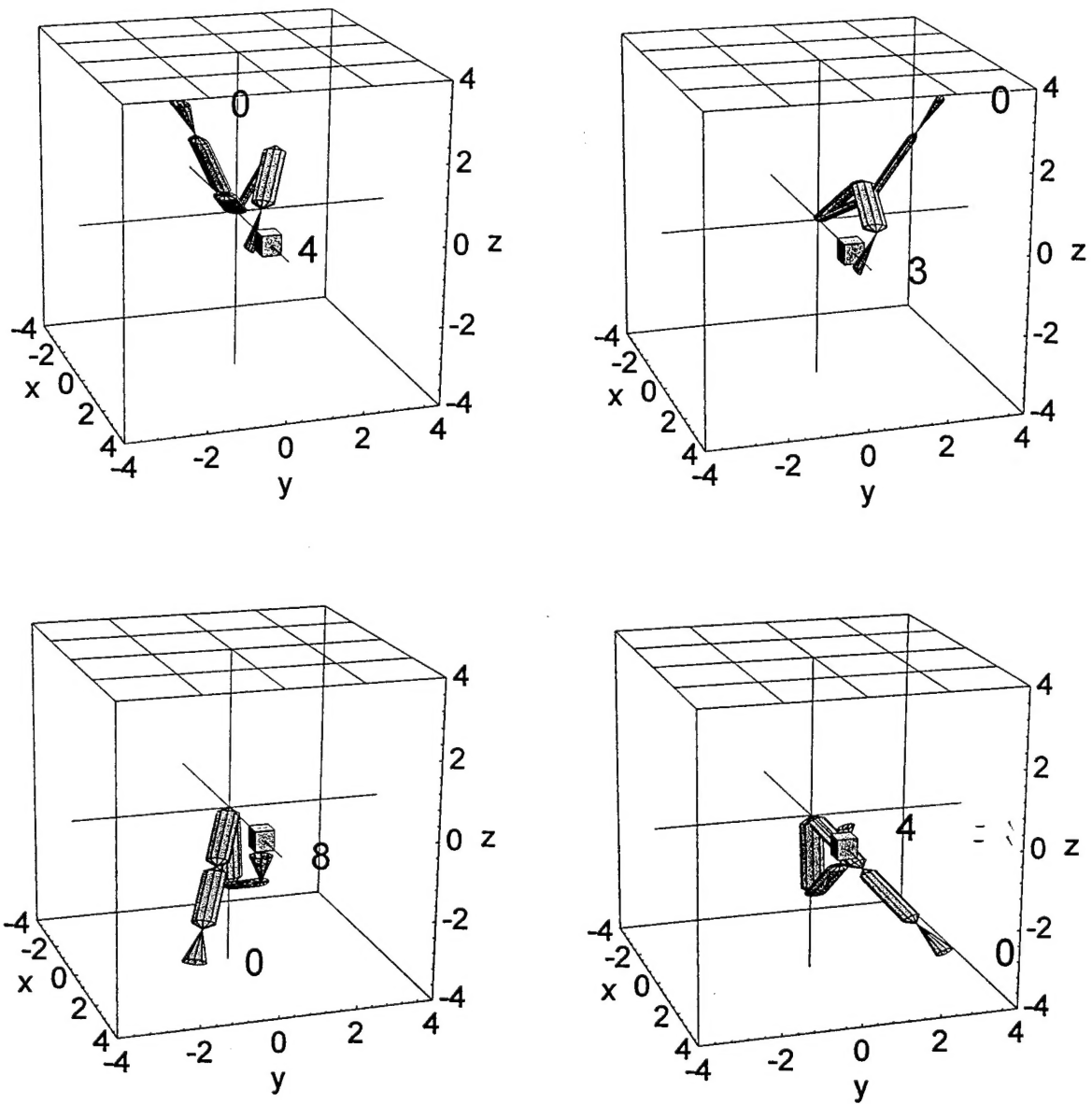Sample of 4 Proximal Targets



6

Figure 1

# Topological Representations of Workspace by Input Weights of Hebbian End Point Controller

Randomized Weights Before Training    Trained in Hemispheric Workspace    Trained in Cubic Workspace



( Filled circles indicate workspace coordinates represented by each neuron; blue lines show topological connections to nearest neighbors in 3D neuron lattice )

7

**Figure 2**

Hebbian Trajectory Controller: Trajectories to Same New Target from Different Initial Postures



( Initial hand position = 0, terminal hand position designated by number of steps to capture target )

Figure 3